Spring Boot

- By Ashok Bollepalli

Ashok IT School

Agenda

- What is Spring Boot
- Why using Spring Boot
- Spring framework Vs Spring Boot
- Pros & Cons of Spring Boot
- Working with Spring Boot
- Spring Starters
- @RestController, @RequestMapping, @Controller
- Customization in Spring Boot

What is Spring Boot ?

- First of all Spring Boot is not a framework, it is a way to ease to create stand-alone application with minimal or zero configurations.
- It is an approach to develop spring based application with very less configuration.

Why Spring Boot ?

- Easy to Create Standalone Spring applications
  - Embedded Tomcat and Jetty Servers directly
- Configure Opinionated ‘starter’ POMs
  - Simplifies Maven configuration
- Automatically Configure Spring whenever possible
- Provides Production ready features
  - metrics, health checks
- Externalization

What Spring Framework Providing ?

What Spring Boot Providing ?

Pros of Spring Boot

- It is very easy to develop Spring Based applications with Java or Groovy.
- It reduces lots of development time and increases productivity.
- It avoids writing lots of boilerplate Code, Annotations and XML Configuration.
- It is very easy to integrate Spring Boot Application with its Spring Ecosystem like Spring JDBC, Spring ORM, Spring Data, Spring Security etc.
- It follows “Opinionated Defaults Configuration” Approach to reduce Developer effort
- It provides Embedded HTTP servers like Tomcat, Jetty etc. to develop and test our web applications very easily.
- It provides CLI (Command Line Interface) tool to develop and test Spring Boot (Java or Groovy) Applications from command prompt very easily and quickly.
- It provides lots of plugins to develop and test Spring Boot Applications very easily using Build Tools like Maven and Gradle
- It provides lots of plugins to work with embedded and in-memory Databases very easily.

Limitations of Spring Boot

- It is very tough and time consuming process to convert existing or legacy Spring Framework projects into Spring Boot Applications. It is applicable only for brand new/Greenfield Spring Projects.

Wrap up

Spring Boot =

Existing Spring Framework + Embedded HTTP Servers – XML or Annotations.

Spring Boot Approaches

There are multiple approaches to create Spring Boot project. We can use any of the following approach to create application.

- Spring Maven Project
- Spring Starter Project Wizard in STS
- Spring Initializer
- Spring Boot CLI

Spring Boot Starters

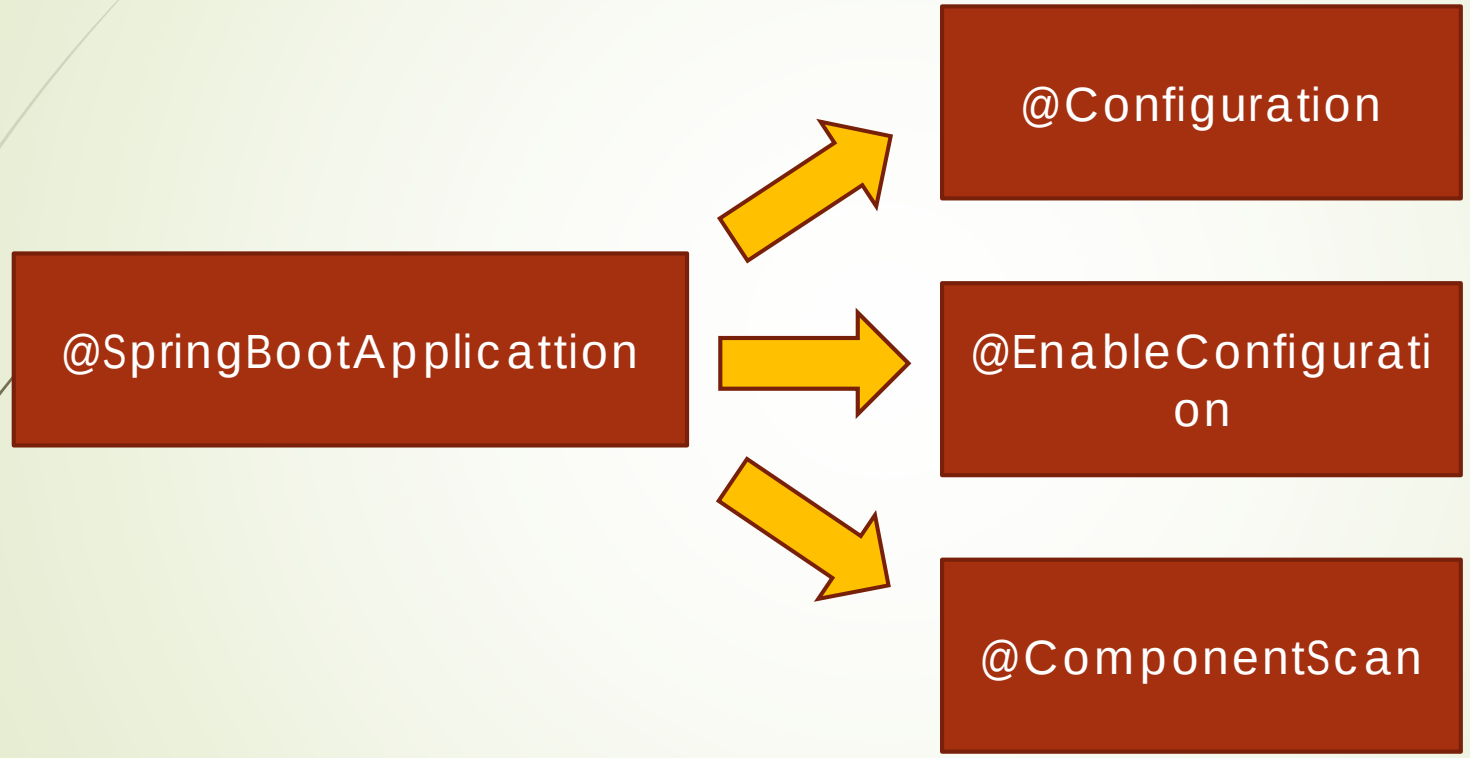
- Spring-boot-starter
- Spring-boot-starter-actuator
- Spring-boot-starter-aop
- Spring-boot-data-jpa
- Spring-boot-starter-security
- Spring-boot-starter-test
- Spring-boot-starter-web
- Spring-boot-starter-thymeleaf
- Etc.

Boot Starter Parent- Maven

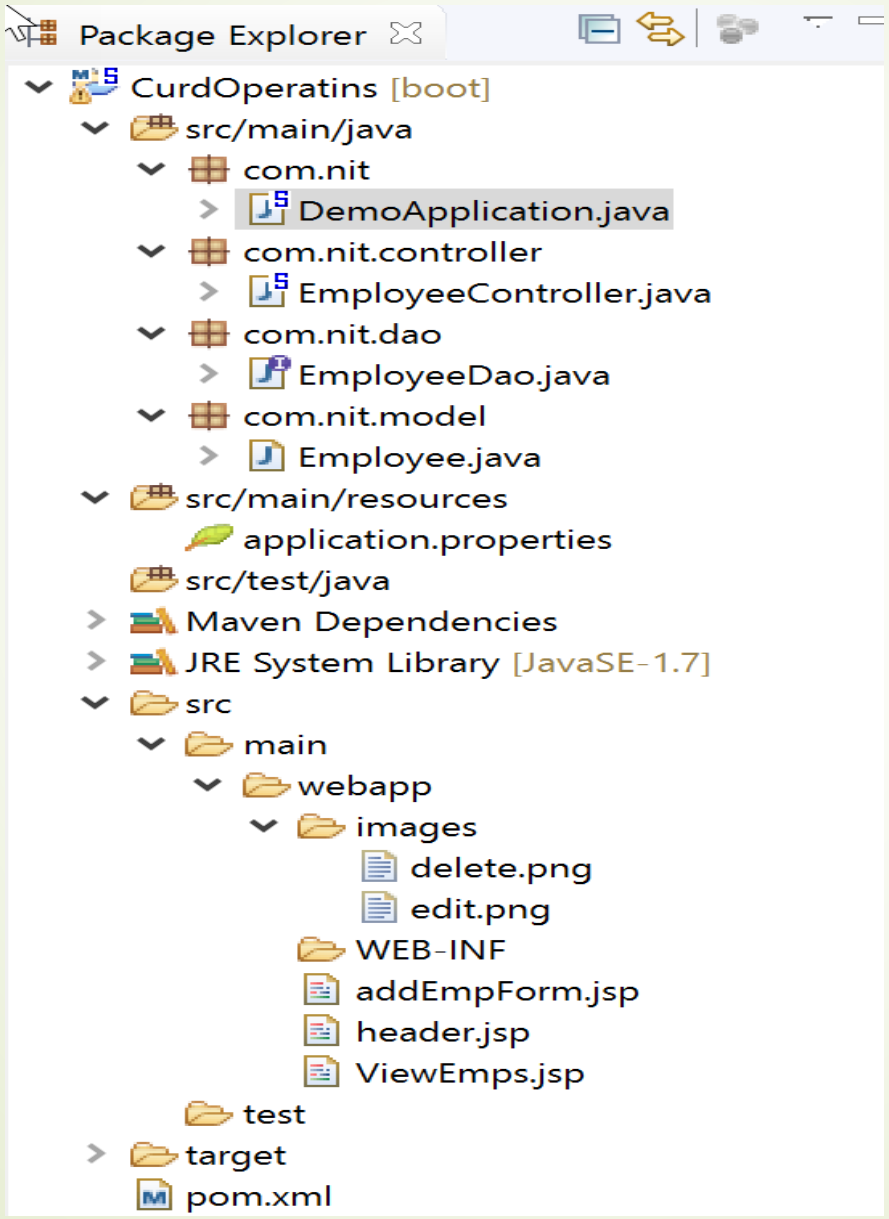
```
<parent>
    <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.7.RELEASE</version>
</parent>
```

Spring Boot Web - Maven

```
<dependency>
    <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```



Spring Boot Project Structure



Ashok IT School

Spring Boot – Boot Strapping class

DemoApplication.java

```java
package com.nit;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

@RestController – to display msg in browser

```
package com.nit.sb.apps.controller;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class WelcomeController {

    @RequestMapping("/welcome")
    public String welcome() {
        return "<h1>Welcome to Spring Boot</h1>";
    }
}
```

@Controller – to return jsp as view

```
@Controller
public class DateController {

    @RequestMapping("/date")
    public String displayDate(Model model) {
        // adding msg to model
        model.addAttribute("msg", new Date());
        // returning view name
        return "dateDisplay";
    }
}
```

Configuration – application.properties

```
#Tomcat Server port number
server.port=9090

#View Details
spring.mvc.view.prefix=/views/
spring.mvc.view.suffix=.jsp

## Db Details
spring.datasource.url=jdbc:oracle:thin:@localhost:1521/xe
spring.datasource.username=system
spring.datasource.password=admin
spring.datasource.driver-class-name=oracle.jdbc.driver.OracleDriver

##ORM Details
spring.jpa.properties.hibernate.show-sql=true
spring.jpa.properties.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.OracleDialect
```

Thankyou..!!

Ashok IT School